ZABBIX SUMMIT ONLINE /2021

DEFINING FLEXIBLE PROBLEM THRESHOLDS WITH THE NEW TRIGGER SYNTAX

SERGEY SIMONENKO

Technical Support Engineer, Zabbix, Latvia

01

NEW SYNTAX AND FUNCTIONS

ONE SYNTAX FOR EVERYTHING

UNIFIED SYNTAX

- **Calculated items** and **triggers** share one syntax:

  avg(/host/system.cpu.util,1h) – this is a calculated item

  avg(/host/system.cpu.util,1h)>25 – a trigger

- Most functions can be used **both** in calculated items and triggers

- No worries: tthe conversion from the old syntax to the new one is performed automatically during upgrade

SMART PARAMETERS

- No longer necessary to pass host and item to every function.

  Only history and prediction functions require them (**always as first parameter**):

  last(/host/item)=“success”

  But: just **time**()>=090000 instead of old ~~{HOST:ITEM}.now()>=090000~~

- **Note:** still necessary to use at least one host/item reference in expression

TIME AND TIME SHIFT

- **Time and time shift** parameters are now one parameter:

  (sec|#num)<:time shift>

- Examples:

1d:now/d	Yesterday
1d:now/d+1d	Today
2d:now/d+1d	Last 2 days
1w:now/w	Last week
1w:now/w+1w	This week

NESTED FUNCTIONS

- Nested functions are now possible, for example:

abs(**last**(/host/item))

length(**find**(/host/item,“pattern”))

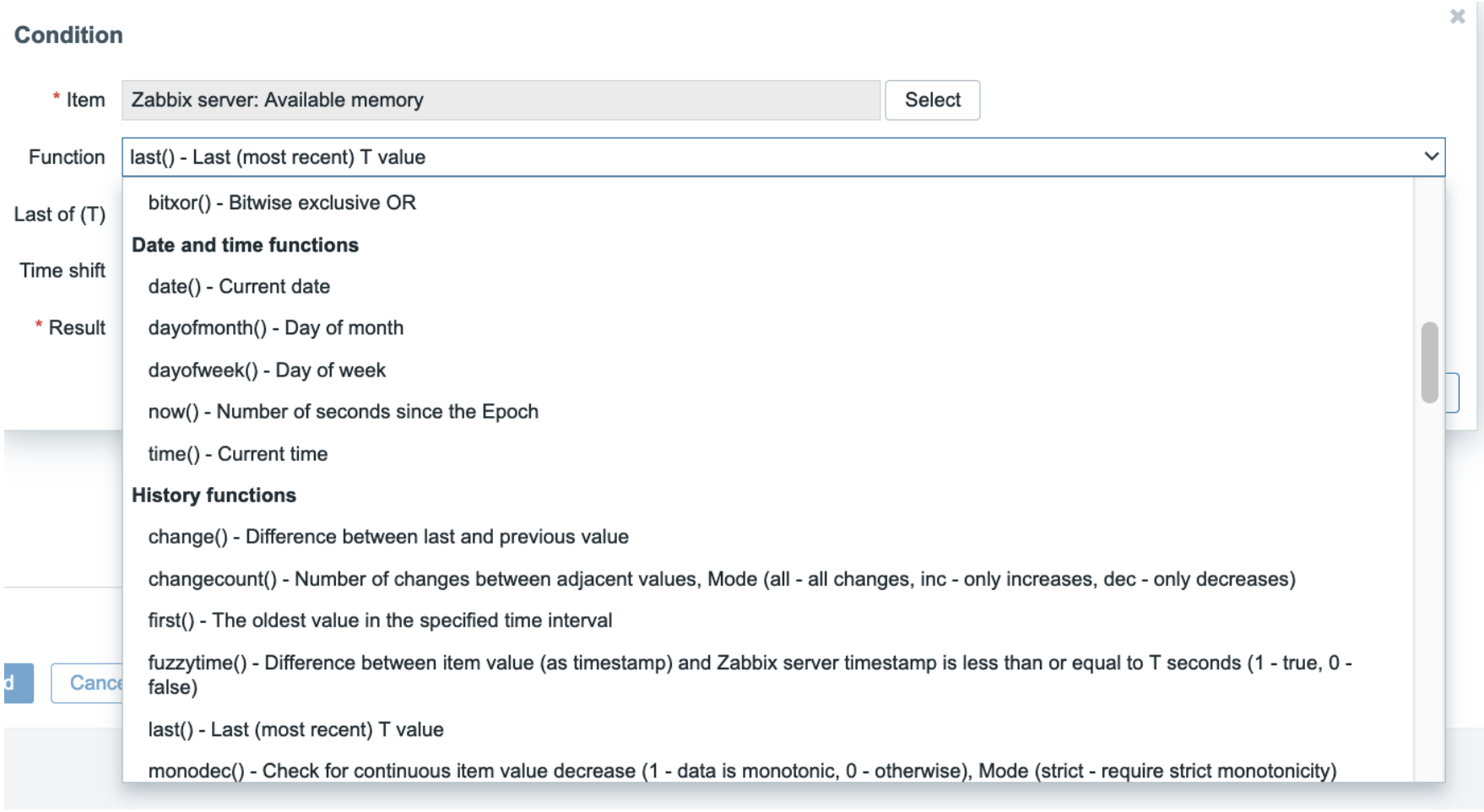
round(**max**(/host/counter,1h))

Redundant functions (~~**abschange**~~, ~~**strlen**~~) were removed

NEW TRIGGER FUNCTIONS

- **History** functions – operate on historical data
- **Aggregate** functions – allow to sum, find minimum and maximum, etc.
- **Operator** functions – enable you to write **compact** and **better** readable expressions
- **Mathematical** functions
- **Date and time** functions (**date**, **now**, **time**, etc.)

NEW TRIGGER FUNCTIONS



NEW STRING AND MATH FUNCTIONS

- **left**, **right**, **mid** – character(s) at given position (index)

  insert, **replace**, **concat**

  trim, **ltrim**, **rtrim**

  ascii, **bitlength**, **bytelength**

  cos, **sin**, **sqrt**, **log**, **mod**, **pi**, **rand** and many more...

OPERATOR FUNCTIONS

- Enable you to write expressions which are **compact** and **better readable**
- Before: ~~{HOST:ITEM.last()}>=1 and {HOST:ITEM.last()}<=10~~

  Now:

  between(**last**(/host/item),1,10)=1
- Before: ~~{HOST:ITEM.last()}=1 or {HOST:ITEM.last()}=2 or {HOST:ITEM.last()}=3…~~

  Now:

  in(**last**(/host/item),1,2,3,…)=1

NEW HISTORY AND AGGREGATE FUNCTIONS

- **monoinc**, **monodec** – detect monotonic increase or decrease in a set of historical values
- **changecount** – count the number of changes (all changes or only increases or decreases) between adjacent historical values
- Functions to easily work with **Prometheus**-compatible emitters data format: **rate**, **bucket_percentile**, **histogram_quantile**

ALSO

- Redundant “shortcut” functions were removed for **easier navigation** and to **avoid confusion**:
- Instead of **delta use:**

  max(/host/item, #100) - **min**(/host/item, #100)
- Instead of diff use:

  last(/host/item) != **last**(/host/item, #2)
- Instead of **prev** use:

  last(/host/item, #2)

02

AGGREGATE CALCULATIONS

REMOVING LIMITATIONS

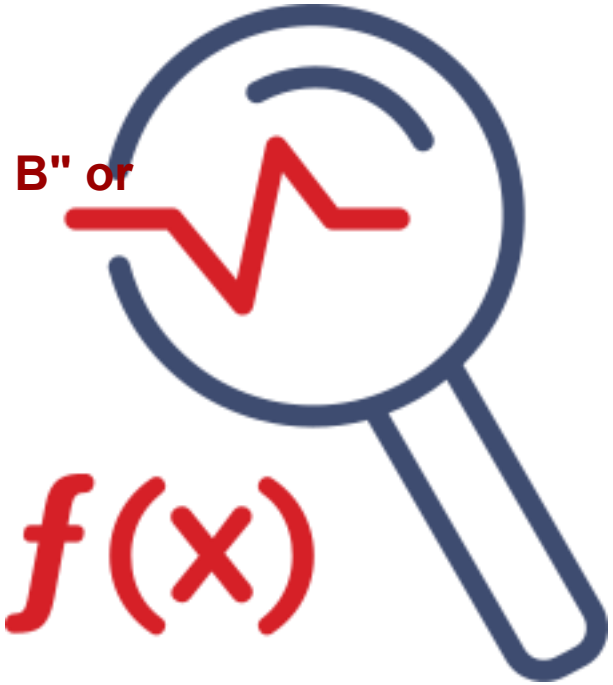
AGGREGATE CALCULATIONS

- Aggregate checks are now part of **calculated items**
- Old syntax only allowed to perform aggregate calculations based on one host group and exact item key:
  ~~grpsum["MySQL Servers","vfs.fs.size[/,total]",last]~~
- **Complex filters** and **wildcards** introduced to address this issue
- This was a **top-voted** feature-request from Zabbix community

AGGREGATE CALCULATIONS

- New syntax is not limited to a single host group for aggregate calculations, you can use **tags**, **multiple hosts groups** and complex **and/or** logical operations with **multiple clauses**.
- Would like to calculate the average CPU load on a certain set of servers?

avg(last_foreach(/*/system.cpu.load**?[group="Servers A" or group="Servers B" or (group="Servers C" and tag="Importance:High")]**))



AGGREGATE CALCULATIONS

- New syntax supports **wildcards** when referencing items in aggregate calculations:
- Want to calculate the total traffic consumed by a customer on multiple hosts and **multiple network interfaces**?

```
sum(last_foreach(/*/net.if.in[*,bytes]?[group="Customer A"]))
```


ZABBIX
SUMMIT
ONLINE /2021
Thank you!
www.zabbix.com